Tuning Latent Dirichlet Allocation Parameters using Ant Colony Optimization

Thanakorn Yarnguy, Wanida Kanarkard
Department of Computer Engineering, Faculty of Engineering,
Khonkaen University, Khonkaen 40002, Thailand.
thanakorn.y@kkumail.com

***Abstract*—Latent Dirichlet Allocation is a famous and commonly used model used to find hidden topic and apply in many text analysis research. To improve the performance of LDA, two Dirichlet prior parameters, namely the α and the *β*,that has an effect on the performance of the system are utilized. Accordingly, they must be set to an appropriate value. Ant colony optimization has the ability to solve the computational problem by adding parameters tuning. Thus, we proposed to implement an approach to find the optimal parameters α and β for LDA by using Ant colony optimization. An evaluation using dataset from the UCI (KOS, NIPS, ENRON) that are the standards for estimating topic model was conducted. The results of the experiment show that LDA, which has tuning parameters by ACO has better performance when it is evaluated by perplexity score.**

***Index Terms*—Ant Colony Optimization; Latent Dirichlet Allocation; Tuning Parameters.**

I. Introduction

There are many data types and one of the most prevalent is the text picture video. Text mining is a common technique used to process the text for the purpose of finding knowledge. There are many methods to represent large text data, such as document clustering, sentiment analysis, an information retrieving. Latent Dirichlet Allocation (LDA) [1] is a modeling algorithm used to analyze large documents. It is based on the main concept that documents consist of topics distributed in the forms of word. LDA has been used to understand hidden semantic of enormous text data. It is used in various text analysis applications, such as analysis of news articles [2] and library application for data search [3]. During the process of using LDA, users must choose a number of topics K and the Dirichlet prior to the parameters, which affect the probability of document-topic distribution. There are two hyper parameters α and β, which are significantly related to model the performance.

A topic modeling can be optimized using various types of optimization algorithm, such as simulate annealing [4-5], genetic algorithm [6] and swarm intelligence [7], which copy the social interaction of animal. Ant colony optimization (ACO) [8] is an algorithm, which has an inspiration from exploring the ability of ants. It is a famous swarm intelligence used for solving optimization problem. ACO has the ability to retrieve an excellent value and produce good result from the problem. Accordingly, this technique is applied to solve optimization problem of LDA. It is applied to tune LDA parameters.

This paper proposed a new method that uses optimization algorithm, namely the Ant colony optimization to find the appropriate values of Latent Dirichlet, which are allocation prior to parameters α and β, wherein they can have effect the performance of the algorithm. The rest of this paper is organized as follows. The literature review and theory are presented in Section II, which includes the description of Ant colony optimization and Latent Dirichlet allocation. Section III presents the methodology, datasets and evaluation method. Meanwhile, Section IV presents the result of the experiment and discussion. Finally, the conclusion is presented in the last section.

II. Literature Review

A standard dimension reduction technique in information retrieval is Latent semantic analysis (LSA) [4], which uses singular value decomposition (SVD) to embed the high dimension space document to low dimension space. It is used to solve the ambiguity of natural language problem and the improvement of LSA is the Probabilistic latent semantic analysis (pLSA) [9]. LSA is a statistical technique used to extract the so-called "topics" for the analysis of co-occurrence data. In 2003, Blei et al. [1] proposed Latent Dirichlet Allocation (LDA), which is used to solve text problem and in text mining application.

A. Latent Dirichlet Allocation (LDA)

LDA is made from the concept of corpus, which consists of many documents and hidden topic distributed in the documents in the form of words. For example, the research article of computer engineering student might contain latent topics related to the concept like Networking, database, wireless sensor and artificial intelligence.

LDA has several processes: For each document d which topics z are probability distributed over word ω. First, we choose a document –specific distribution over topics P($z|d$).

$$P(z|d) \sim \mathrm{Dir}(\alpha) \quad (1)$$

Then, for each ω, we choose a topic at random according to this probability.

$$P(\omega|z) \sim \mathrm{Dir}(\beta) \quad (2)$$

where the hyper parameters for the Dirichlet priors are α and β.The probability of ω was:

$$P(\omega|d,\alpha,\beta) = \sum_{z=1}^{Z} P(\omega|z,\beta)P(z|d,\alpha) \quad (3)$$

where Z is a number of latent topics.

There are many optimization algorithms for the improvement of the performance of topic modeling

algorithm. For example, Genetic Algorithms (GAs) is a technique based on the theory of selection and evolution, which are adaptive heuristic search algorithm. Panichella and teams [6] used LDA to process data in the field of software engineering. They applied genetic algorithm (GA) to adjust LDA parameters. It consists of hidden topics (K) parameters, Gibb sampling iteration, the distribution of the topic in document (α) and the distribution of term in topic(β). The process of GA-LDA increased the parameters value followed by the software engineering task. Simulated annealing (SA) is a probabilistic technique, which applies thermo dynamic process to solve optimization task. Andrey Kuzmenko [11] introduced simulate annealing (SA) to adjust LDA parameters. The distribution of topic in document (α) has a value not over 1 and has been adjusted in burn-in process α:1.0→0.001. Swarm intelligence, which consists of a population interacting with each other and their environment such as birds, ants, fish. Latha and Rajaram [12] has improved the performance of Latent semantic index using Particle swarm optimization (PSO) and used k-mean to process the concept document matrix and used simulate annealing to process term document matrix. Hasanpour [5] has applied Particle swarm optimization with latent semantic indexing to clustering document. PSO has used to change weight and this method has been used to reduce document dimension. The result showed that clustering performance has improved.

B. Ant Colony Optimization (ACO)

ACO was introduced in the 1990s. It is a meta heuristic which based on the behavior of ant in nature. Ants are seeking for food by random walk to find the best route and have the ability to drop pheromone on selected route. Pheromone which drops on the path will help other ants find the best way from the nest to the target resource and the amount of pheromone on route will decay over time. ACO was applied to solve the optimization problem.

Algorithm ACO
Initial ants parameters, create pheromone trails
while expiry condition not met do
Each Ant built Solutions
Update Pheromones on trails
End while

Figure 2: ACO algorithm

The solution of the system can be identified by the probability that changes the pheromone value (τ). Defined ant(m) to solve the problem, pheromone value τ_{ij} which relate to edge from i to j are adapted by:

$$\tau_{ij} \leftarrow (1-\rho)\cdot\tau_{ij} + \sum_{k=1}^{m}\Delta\tau_{ij}^{k} \quad (4)$$

when ρ is a pheromone evaporation rate and $\Delta\tau_{ij}^{k}$ is the pheromone value on the edge i to j.

$$\Delta\tau_{ij}^{k} = \begin{cases} \frac{Q}{L_k} & if\ ant(\mathrm{k})\ choosen\ edge\ (i,j) \\ 0 & otherwise \end{cases} \quad (5)$$

Q is a constant and L_k is an edge which Ant(k) has built. Ant will choose the node by Stochastic mechanism. In this case, the probability to walk from node i to j is:

$$p_{ij}^{k} = \begin{cases} \frac{\tau_{ij}^{\alpha}\cdot n_{ij}^{\beta}}{\sum_{c_{il}\epsilon N(s^p)}\tau_{il}^{\alpha}\cdot n_{il}^{\beta}} & if\ c_{il}\epsilon N(s^p) \\ 0 & otherwise \end{cases} \quad (6)$$

when $N(s^p)$ is a set of the probability of edge (i,l) when l is a node which ant(k) never visit and parameters α and β influence the pheromone trail.

Ant colony optimization (ACO) is used to optimize the parameters of the algorithm to retrieve good performance. Zhang, XiaoLi, et al. [13] used ACO optimizing support vector machine parameters selection C and σ. They found the approach produce less calculation time and high accuracy compared to other methods. Alwan et al. [10] used ACO to adjust SVM parameters C and γ. They used 8 datasets from UCI and the result showed that ACO-SVM provides good accuracy compared to another method such as PSO-SVM and GA-SVM.

III. Methodology

A. Proposed Algorithm

This study uses Ant colony optimization to optimize Latent Dirichlet Allocation parameters. ACO is used to optimize LDA parameters α and β, which are divided within the range and each ant will use probability to choose a value and generated the result. The process of ACO has started with initial ants. The number of ants will be set and the path between ants to the food resource will be created. When ants are traveling between the source and destination, some pheromones are dropped on the trail and this will decay over time. Ants select the path to travel according to a stochastic mechanism. In this paper, we did not fix the amount of pheromone, but it was selected randomly on every iteration. This approach leads to a rapid or slow reduction of the pheromone value in some cases. ACO is an iteration algorithm. The number of iteration has been set in the initial phase.

In addition to evaluation of LDA and ACO-LDA, we also evaluated the proposed method in term of perplexity, which is a standard for estimating topic model performance. The perplexity of a set of word, $(W_d|a_d)$ for d is a member of corpus D, which is defined as Equation (7).

$$perplexity(W_d|a_d) = \exp\left[-\frac{\ln\rho(W_d|a_d)}{N_d}\right] \quad (7)$$

The process of an algorithm is shown in Figure 1. The main steps were preparing datasets, initial ACO, running LDA, Update pheromone and Sorting perplexity.

Dataset was loaded to memory at the beginning of the process. The experiment used 4 bags of word datasets from UCI machine learning repository. Datasets have already ben tokenization, removed the stop word and truncated the vocabulary, which occurs less than ten times. To evaluate the text analysis experiment, these data sets are used wildly in clustering and classification task. Datasets consist of technical papers from the NIPS – the collection of research papers contributed by researchers on learning algorithms field, Enron-The CALO Project (A Cognitive Assistant that Learns and Organizes) was collected in this dataset, which contain data from around a hundred of users, mostly the member of Enron and KOS- The collection of blog entries. The statistics of datasets are represented in Table 1. D is a

number of documents and N is the number of tokens of the documents and W is the vocabulary size.

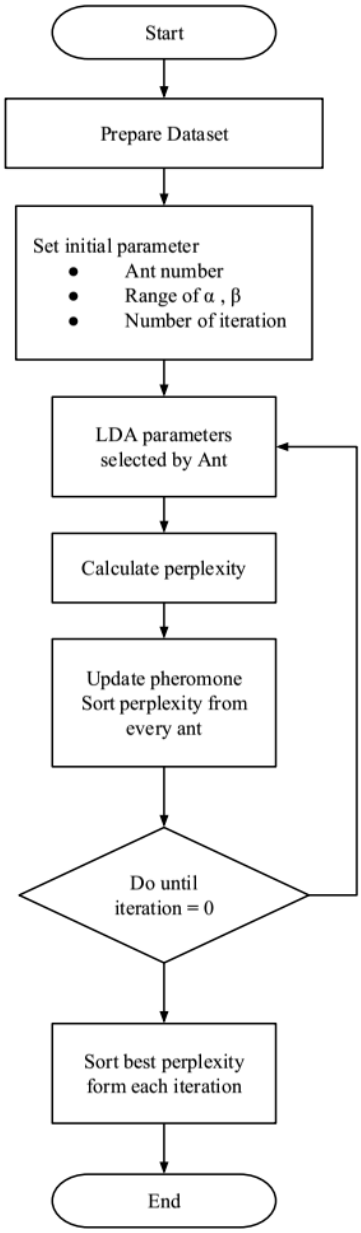

Figure 1: Proposed ACO-LDA algorithm

Table 1
Datasets Statistics

dataset	D	W	N
Kos	3430	6906	467714
Nips	1500	12419	1,900,000 (approx)
Enron	39861	28102	6,400,000 (approx)

Next, the parameters of ACO will be initialed, which include the number of ant and the number of iteration. In this step, LDA parameters α and β are divided. For each iteration, the value of Pheromone Constant and Decay Constant are randomly selected, and each ant will select the path and get α and β value and introduce it to LDA. After computing the LDA, Perplexity score is calculated for each selected parameter from each ant and sorted to find the minimum value of each iteration. The pheromone on the path will be updated according to the value of Pheromone Constant. In the next iteration, some pheromone will decay depending on the Decay Constant. When the maximum number of iteration are archived, the minimum perplexity of each iteration will be sorted to find the best performance.

IV. Results and Discussions

We also run LDA by topic modeling toolkit –Gensim, which includes much open-source text analysis library implemented by Python language and customized by python programing to run ACO. In this process, the comparison of LDA and ACO-LDA are measured by Perplexity. The initial parameters setting of LDA is the difference of each datasets. The experiments were performed on a Desktop which uses intel CPU Core(TM) i7-4700HQ at clock speed 2.40 GHz with 12.0 GB of RAM and 64-bit operating system.

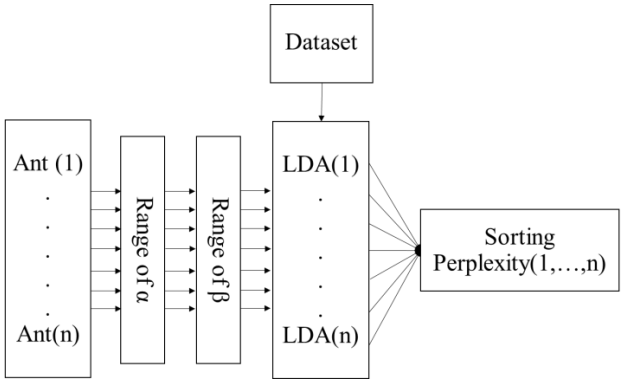

Figure 2: Diagram of ACO-LDA

First, we used KOS dataset to evaluate LDA, the number of the topic is K=10 and the value of parameters α to 2/K and β = 0.01 are set, while the LDA ran for 120 times. ACO-LDA parameters are initialized as follow: the number of ants = 10, iterations = 12, Pheromone Constant and Decay Constant are randomized in the range (0.1-1.0), which the Decay constant must not be over the Pheromone Constant. Parameters of LDA are set as a range, wherein α is between 0.005-0.25 and β is between 0.0025-0.03. The results obtained from the experiment is shown in a graph in Figure 3.

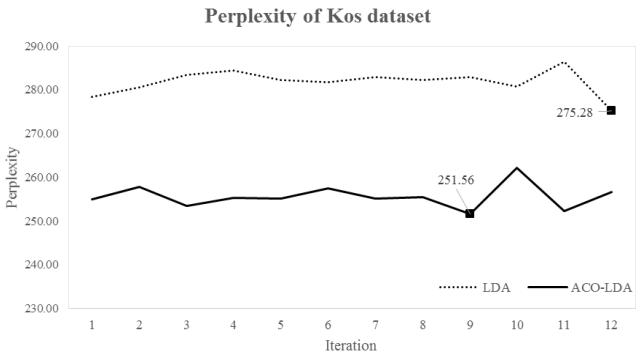

Figure 3: Perplexity comparison of KOS datasets which define K=10

From Figure 3, we plot the best perplexity of LDA from every 10 values. ACO-LDA is running for 12 iterations and we plot only the best perplexity from 10 ants of each iterations. The result from LDA and ACO-LDA when using KOS dataset showed that the perplexity of ACO-LDA is less than LDA.

Secondly, we run the LDA on Nips datasets which is larger

than KOS and set the number of topic K= 30, α is 2/20 and β = 0.01 and ACO-LDA is the adjusting range of α to (0.025-0.2) and β to (0.0025-0.3). The result showed that the perplexity of ACO-LDA is less than LDA as shown in Figure 4.

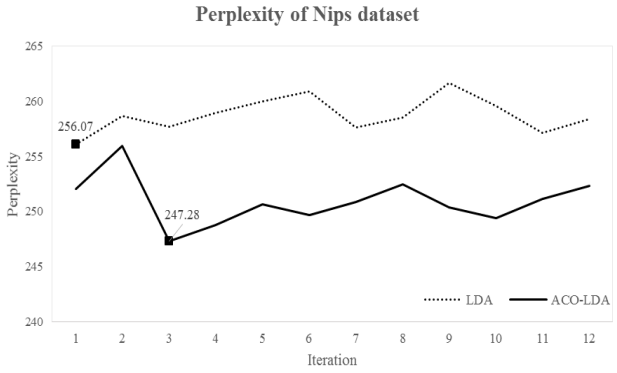

Figure 4: Perplexity comparison of Nips datasets which define K=30

Third, Enron datasets were processed by LDA and adjusted a number of the topic to K= 100. Distributed parameters α is set to 0.02. The parameters of ACO-LDA is set in a range (0.025-0.2) for α and β is set in range (0.0025-0.03). The result of the experiment was shown in Figure 5. The perplexity of ACO-LDA was 326.89 and the perplexity of LDA was 339.20.

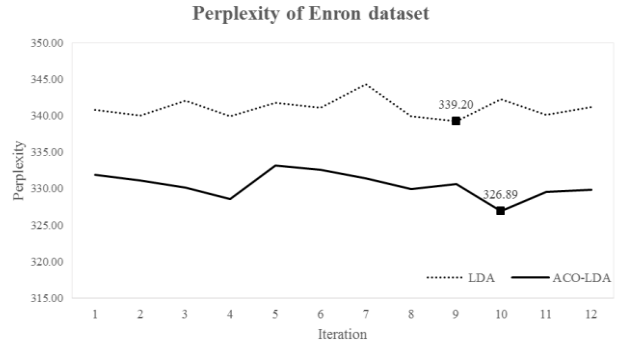

Figure 5: Perplexity comparison of Enron datasets which define K=100

To evaluate the performance of LDA and ACO-LDA, we used perplexity score, the less value of perplexity means better performance of the algorithm. The result from Figure 3, 4 and 5 showed that the perplexity of ACO-LDA is less than LDA in KOS, NIPS and ENRON dataset. When ACO was used to find parameters of LDA. It can get better performance because ACO was systematically randomize the value of LDA parameters by ants.

V. CONCLUSION

This paper proposed a technique using Latent Dirichlet allocation (LDA) to find hidden topic from the standard dataset for topic modeling evaluate provided by UCI. We choose 3 datasets (KOS, ENRON, NIPS). The performance of LDA can be improved by adjusting hyper parameters α - the parameter of the Dirichlet prior on the per-document topic distributions and β - the parameter of the Dirichlet prior on the per-topic word distribution. Ant colony optimization has been using to tune LDA parameters α and β. ACO is initial Ants to find the best parameters of LDA. Pheromone Constant and Decay Constant are randomly selected. In each of the iterations, ant has selected the parameters based on the pheromone value on the path. To evaluate this method, we compared the perplexity of LDA and ACO-LDA. The evaluation results indicated that ants could find the appropriate value for LDA and ACO-LDA with tuning parameters by ACO has better perplexity than LDA.

ACKNOWLEDGMENT

The authors wish to thank Kalasin University for financial support this research and colleagues from Department of Computer Engineering.

REFERENCES

[1] Blei, David M., Andrew Y. Ng, and Michael I. Jordan. "Latent dirichlet allocation." *Journal of machine Learning research* 3.Jan (2003): 993-1022.

[2] Newman, David, et al. "Analyzing entities and topics in news articles using statistical topic models." *Intelligence and Security Informatics* (2006): 93-104.

[3] Mimno, David, and Andrew McCallum. "Organizing the OCA: learning faceted subjects from a library of digital books." *Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries*. ACM, 2007.

[4] Dumais, Susan T. "Latent semantic analysis." Annual review of information science and technology 38.1 (2004): 188-230.

[5] Hasanpour, E. H. "PSO algorithm for text clustering based on latent semantic indexing." *The Fourth Iran Data Mining Conference*. Tehran, Iran. 2010.

[6] Panichella, Annibale, et al. "How to effectively use topic models for software engineering tasks? an approach based on genetic algorithms." *Proceedings of the 2013 International Conference on Software Engineering*. IEEE Press, 2013.

[7] Liu, Yang, and Kevin M. Passino. "Swarm intelligence: Literature overview." Department of Electrical Engineering, the Ohio State University (2000).

[8] Dorigo, Marco, and Gianni Di Caro. "Ant colony optimization: a new meta-heuristic." Evolutionary Computation, 1999. CEC 99. *Proceedings of the 1999 Congress on. Vol. 2*. IEEE, 1999.

[9] Hofmann, Thomas. "Probabilistic latent semantic indexing." *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 1999.

[10] Alwan, Hiba Basim, and Ku Ruhana Ku-Mahamud. "Solving SVM model selection problem using ACOR and IACOR." *WSEAS Transactions on Computers*: 277-288.

[11] Kuzmenko, Andrey. "Simulated Annealing for Dirichlet Priors in LDA." (2014).

[12] Latha, K., and R. Rajaram. "An Efficient LSI based Information Retrieval Framework using Particle swarm optimization and simulated annealing approach." *Advanced Computing and Communications, 2008. ADCOM 2008. 16th International Conference on. IEEE*, 2008.

[13] Zhang, XiaoLi, et al. "A grid-based ACO algorithm for parameters optimization in support vector machines." *Granular Computing, 2008. GrC 2008. IEEE International Conference on. IEEE*, 2008.